Use Case/Requirements Template. Version 1.2

Step 1. Write High Level Use Case
Step 2. Write initial requirements document
Step 3. Write Expanded Use Case (stored on same page with High Level Use case)
Step 4. Refine Requirements if necessary.

Tool ID. Use Case Name (should start with a verb, e.g., Read Event Data.)

High Level Use Case

Actors: Examples are GI or external component
Goal: What the actor is trying to achieve (a 1 line summary)
Trigger: What starts the use case. (Usually an action by the main actor, but could be some timed event).
Description: Terse paragraph describing what happens. E.g., The program reads the user specified data file. The paragraph should not include details like file format (FITS data file) unless this has been decided as an external requirement on the file format.)
References: list of requirements that this use case addresses based on the numbers of the headings in the requirements document: 1.1, 2.1, etc. If the requirement comes from a different tool or a general requirements document then there will be a prefix for the requirement number which indicates where the requirement comes from. E.g., U14.1.1 means that it is requirement 1.1 form the U14 tool document. (These references will be added when requirements are written; the next step after the high level use cases.)
Use Case Template version: 1.2

Expanded Use Case

(This is to be done after the High Level Use Cases and rudimentary requirements are written.) I have put the expanded use case template after the high level use case because they should be stored together (on the same page in the document we finally produce).

Preconditions: What must be true in order for the action to begin
Successful End: What happens if the use is a success (e.g., data read into program).
Failed End: What happens if the use case utterly fails and alternative courses do not resolve the problem (e.g., program terminates.)
Priority: How important this use is to the system. (levels 1-3: 1 means critical for proper tool operation, 3 means, just a nice to have feature)

Typical Course of Events

	Actor Action		System Response
1	GI specifies file name on command line	2	read filename from CL
		3	Open file.
		4	read data in file

		A1	When alternate courses are developed, we can expand on the sequence here.,
A2	Alternate course causes actor to do something		
		B1	

Alternative Courses
A. Line #: e.g. error conditions, optional courses, like “Line 3:
File not found, prompt user for new file.”
B Line #: A separate sequence started by a different alternative course.

Blank Template follows on next page.

Tool ID. Use Case Name

High Level Use Case

Actors:
Goal:
Trigger:
Description:
References:
Use Case Template version: 1.1

Expanded Use Case

Preconditions:
Successful End:
Failed End:
Priority:

Typical Course of Events

	Actor Action		System Response

Alternative Courses
A1.

ToolID Requirements

• Overview Statement

A short description of the purpose of the tool. The purpose of this too is to do something clever with the data and write out results. E.g., Find optimal gravitational lens shape and mass to explain multiple images of GRB source.

• Goals

These include the Goals from the use cases:

- Calculating the Ricci tensor that is saved to a file.
- Etc.

• Requirements List (A list of requirements for the tool)

1. Input Parameters

1.1. Get input parameters for the program such as name of event data file, and names of algorithms

1.2. etc.

2. Message services

2.1.